Common FEP Output Format Issue Resolution Team
Minutes of 4th Meeting (Revised)
May 8, 1996 2:00-3:30 pm Room 91

Attendance

- Ray Deyton (System Mon)
- Dave Fish (System Mon)
- Colin Furtaw (M/W)
- Mike Garvis (M/W, FEP)
- Ed Greville (facilitator)
- Lucas Huang (M/W)
- Farshid Khoui (Data Mgt)
- Cara Lilley (M/W)
- Dennis Sasaki (FEP)

1) Lucas Huang presented two options for the Common FEP Output Format packet definition in a handout entitled "Options for FEP Output Telemetry Data Distribution Objects (API)". These options correspond to Options A (time boundary) and B (minor frame) under Issue #2 in the issues table below. Option A was selected as the more favorable. An update to this handout, containing changes agreed to in the meeting, is included as an attachment. This specification for the FEP Output Common Data Format may still evolve as the design of the FEP and Data Management progress.

2) A systematic review was conducted of all the issues and questions collected during the previous meetings. The table below contains the decision made regarding each of these issues at present.

3) It was generally recognized that the engineering implications of format decisions (i.e. performance, storage, network loads, etc.) are important considerations and that more work must be done in this area. It was agreed that prototyping is a more effective way to size the data rate for the decommutated and EU-converted telemetry stream than analysis. A working prototype is planned to be operating by July 1, 1996.

COMMON FEP OUTPUT FORMAT IRT ISSUES

Issue # 1 -- Timing for Calculation of Context Dependent Parameters

Option A: Output raw value, then wait to compute EU until both values have been received. Different time stamps will be assigned to raw and EU values.

Option B: Compute both raw and EU at the same time based upon preceding indicator--both will receive same time tag--boundary conditions can potentially lead to incorrect values

Approach: Option B will be followed. This matches the design of current system.

Issue # 2 -- Definition of an FEP Packet

Option A: Time boundary. All data that arrives within a given time span is provided in a single packet. Will result in packets being generated at a 40 Hz. rate based on telemetry format. If packet is too big to fit in FEP packet size limitation, two or more adjacent packets will contain the same time stamp. Additional packets will slightly increase the network loading.

Option B: Minor frame boundary. Each packet contains one complete minor frame. If this option is selected, then time-tagging issues for "super com" data must be addressed.

Option C: Hardware packet boundary. One packet contains the maximum amount of data the FEP can ship in a single packet. Size is optimized based on FEP configuration. (This approach requires more work for the data merging process.) This option will not be selected since it is more appropriate to let software processing drive the size.

Approach: Option A avoids the issue of time-tagging "super com" data and is acceptable to the merge design team. Option B is most favored by the merge design team, however, Option A is only slightly less desirable. Option C is least desirable.

Issue # 3 -- Time Stamps Required in Packets Provided by the FEP

Option A: Spacecraft time.

Option B: Spacecraft and CCS processing time.

Option C: Spacecraft and GRT.

Option D: Spacecraft, GRT, and CCS processing time

Approach: Option D was selected. This was advocated by the Time-Tagging Issue Resolution Team for the following reasons. Spacecraft time is needed by users of the data. GRT and CCS Processing Time are needed to perform final time validation and data merging. Since data merging is done in Data Management, and since we want to minimize re-reading the raw data from the FEP short-term storage, GRT and CCS Processing Time must be sent to Data Management.

Issue # 4 -- Location of Time Stamps Within FEP Packets

Option A: Time stamps are contained once within each packet. This option is only available if Option A is selection for Issue #2 because that option defines a packet by a specific time. This is the most efficient solution from a data volume perspective.

Option B: Time stamps are contained once within each packet element (once for each parameter). This option must be selected if Option B is selection for issue #2 because, with super commutated data, more than a single time would be present in each packet.

Approach: Option A has been selected.

Issue #5 -- Element Definition

Option A: Each element contains both raw and EU values. Provides most efficient space and network utilization at the risk of boundary condition errors related to context-dependent EU conversion.

Option B: Separate elements are provided for raw and EU values. Provides opportunity for more accurate context-dependent EU conversion at the expense of increased network traffic and the need for separate identifiers for raw and EU versions of a mnemonic’s value.

Approach: Option A has been selected.

Question: Do we really know that Option A gives the lower data volume? Consider that many of the parameters don't require EU conversion? With Option A, every parameter requires an EU field even if the value is blank.

Issue # 6 -- Location and Number of Flags

Flags will be provided for each element. Since the resolution to Issue 5 is to combine raw and EU values in the same element, the flags will be sent once for each telemetry value received. Two bytes will be reserved for flag values.

Issue # 7 -- Handling of bad quality data

Option A: Forward all recognizable data through the FEP to Data Management. Quality flags will be used to indicate data with questionable time tags or other problems.

Option B: Forward only what looks like good data. Data Management would have to request segments from the FEP short-term raw storage to fill gaps during data merge operations.

Approach: Option A has been selected. Since we are providing all three time stamps (Spacecraft Time, CCS Processing time and GRT), Data Management will have everything it needs to perform the data merge even when data anomalies are present.

Issue # 8 -- Handling of OBC dumps

This issue has been deferred since dumps are not a Release 1 requirement.

Question: Can we afford to do this since Release 1 is the easy release? Maybe we'd better settle some of the basic issues now and avoid the rush between Releases 1 and 2.

Issue # 9 -- Simultaneous Handling of Multiple Data Streams

The interface from the FEP to Data Management must be able to handle at least three simultaneous data streams. These include Real-Time telemetry; ETR, SSR, LOR, and FEP playback data; and OBC dumps. Each stream will be delivered to the archive via a unique socket.

Issue # 10 -- Format of FEP Playback Data from 30-day "Raw" Archive

The packet and element structure for data produced by playing back raw short-term archived telemetry data is identical to normal operation. The playback data source indication is clearly distinguished from the real time source indication. Also playback is run in an off-line mode.

Issue # 11 -- Primary Key for Mnemonics

Each PRD mnemonic is assigned a single, unique primary key that will be preserved for the life of the mission. The current system provides a unique identifier (the UFM) that serves the role of a consistent reference regardless of the specific software application being run or the database in use. CCS members have decided to use a numeric index instead of the UFM to reduce the number of bytes contained in the messages; however, no one has yet mandated that the numeric ID is required to be unique and to be consistent across all applications. Within the monitoring and analysis applications, as well as the FEP, we will also use numeric references for processing efficiency. Instead of each application managing its own set of references, it is more efficient, from an overall system perspective, if one numeric reference is assigned to each mnemonic and is used for the remainder of the mission.

Issue # 12 -- ETR Data Format
ETR/SSR playback data will leave the FEP in forward (not reversed) order, will be terminated with an end-of-data marker, and will be transmitted at a rate throttled by Data Management.

Issue # 13 -- Telemetry Format Changes
A unique identifier will be added to the PRD that corresponds to the telemetry format. Whenever the format changes, an element will be included in the packet identifying the new format.

Issue # 14 -- PDB Version
Upon establishment of a session, the PDB version will be communicated through a separate packet containing a single element identifying the PDB version. Note that changing the PDB version will require terminating the current session so PDB version changes will never happen during a session.

Issue # 15 -- Data Precision for EU Values
EU-converted data will be represented in 32-bit floating point representation.

Actions Items

1) Lucas Huang will update the description of Telemetry Distribution Objects [see attachment].

2) Ed Greville will report on issues related to time-tagging [(done].

3) Dennis Sasaki will find out what triggers the FEP to transfer data to Data Management and System Mon, and how large (in bytes) packets can be and specifically whether the packet size can be increased past 1024 bytes to contain an entire minor frame. [Packet size can be increased. Triggering of data transfers can be programmed.]

4) Ed Greville will find out if Mike Garvis has done a data volume estimate for the FEP output stream. [Approximations have been done. A prototype will be available July 1, 1996]

5) Ed Greville will find out to what extent the 40 Hz attitude telemetry parameters are accurately timed (i.e. really 25 ms apart). [For purposes of Attitude Determination and Sensor Calibration, precise timing is not considered important. It is simply assumed that the 40 Hz data points are equally spaced.]

Additional Issues

1) Data Source Indicator -- The Data Source indication may require multiple fields. Data may originate from real-time, recorder playback. Then, it may be replayed from a ground site such as the TGT LOR. Subsequent to this, it may be replayed from the FEP. Also, it may be simulation or test data either being received from an external simulator or being replayed from the FEP.

2) Format of Time Stamp -- The precise format expressing time must be established. If millisecond accuracy is required over the mission life, at least 40 bits are required.

3) <u>Data Volume Reduction</u> -- There are significant opportunities for reducing the total data volume. These will definitely be applied prior to on-line permanent storage. There may be significant advantages to reducing the data volume at earlier points in the telemetry data flow.

4) <u>Relation to Other Formats</u> -- There are several other significant telemetry formats in CCS besides the FEP Output Common Data Format: data query presentation format(s), FEP raw storage format, Long-term Archive format, and on-line repository format. What are these formats? Are they all different? What is the relationship between them?

Next Meeting: Currently, this work is being planned for continuation under PAT Team design efforts. Further meetings will be scheduled if necessary.

Attachment 1
Options for FEP Output Telemetry Data Distribution Objects (API)

(5/14/96)

Overview:

Telemetry will be distributed by the FEP via telemetry packet objects. The reason for packaging telemetry in objects is to isolate the recipients from format and timing concerns. The telemetry objects will contain all the necessary methods for extracting the data and converting it to the proper format.

Each telemetry object will contain a single telemetry data packet. Each packet will contain meta-data about the packet as well as telemetry element objects. At the lowest level, telemetry data is encapsulated in element objects, the element objects are in turn encapsulated in packet objects, and once packets are completed, they are shipped to the requesting process via middleware. There are two possible architectures (time sequenced and minor-frame sequenced) for the Telemetry Data Objects are as follows.

OPTION A -- TIME SEQUENCED

Telemetry Data contains:	Bit Size	Packet Objects
Packet Object contains:	8	Data Source (multiple sources, non-exclusive)
	8	Data Format
	96	Packet Time-Tag (SCT, UTC, GRT)
	8	Number of Elements in the packet Array of Telemetry Element Objects
Element Object contains:	16	Mnemonic ID
	32	Raw Value
	32	EU Value
	16	Flags

Estimated data expansion over raw data: 3-4

API for end-users is defined as follows:

Retrieve Element Info:	getID()	- returns Mnemonic ID of the element
	getRaw()	- returns raw value of the element
	getEU()	- returns EU converted value or raw if there is no EU
	getFlags()	- returns all flags related to the element
	dump(...)	- returns a preformated dump of data (overloaded)

Retrieve Packet Info:	getSource()	- returns the data source of the packet
	getFormat()	- returns the data format of the packet
	getTime()	- returns the time tag of the packet
	getNum()	- returns the number of elements in the packet
	getElements()	- returns the start location of the element
	dump(...)	- returns a preformated dump of packet (overloaded)

Advantages

- More efficient element information/packet structure - least amount of data duplication

Disadvantages

- Loss of Minor Frame organization may complicate merge process

OPTION B -- MINOR FRAME - SEQUENCED

Telemetry Data contains:	Bit Size	Packet Objects
Packet Object contains:	8	Data Source (multiple sources, non-exclusive)
	8	Data Format
	32	Packet Minor Frame Count
	8	Number of Elements in the packet Array of Telemetry Element Objects
Element Object contains:	16	Mnemonic ID
	32	Raw Value
	32	EU Value
	96	Time Tag (SCT, UTC, GRT)
	16	Flags

Estimated Data Expansion over raw format: 6-7

API for end-users is defined as follows:

Retrieve Element Info:	getID()	- returns Mnemonic ID of the element
	getRaw()	- returns raw value of the element
	getEU()	- returns EU converted value or raw if there is no EU
	getFlags()	- returns all flags related to the element
	getTime()	- returns the time tag of the element
	dump(...)	- returns a preformated dump of data (overloaded)

Retrieve Packet Info:	getSource()	- returns the data source of the packet
	getFormat()	- returns the data format of the packet
	getCount()	- returns the Minor Frame Count of the packet
	getNum()	- returns the number of elements in the packet
	getElements()	- returns the start location of the element
	dump(...)	- returns a preformated dump of packet (overloaded)

<u>Advantages</u>

- Minor Frame organization may simplify the merge process

<u>Disadvantages</u>

- A packet contains elements with different times (40 Hz supercom)
- Time is provided for every element resulting in larger data volume